

МОДЕЛИРОВАНИЕ РАБОТЫ МАНИПУЛЯЦИОННОГО РОБОТА В ПРОГРАММНОМ ПАКЕТЕ MATLAB ROBOTICS TOOLBOX

Е.Е. Егоров

egorov.18011996@gmail.com

SPIN-код: 2333-2610

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Исследованы возможности программного пакета MATLAB Robotics Toolbox для моделирования движения манипуляционного робота. Описан алгоритм создания модели манипулятора в программе на основе представления Денавита — Хартенберга. Проведено исследование кинематики манипуляционного робота. Решены прямая и обратная позиционные задачи при помощи внутренних функций программного пакета. Рассмотрено создание приложения с графическим интерфейсом в среде GUIDE для отрисовки модели робота и моделирования его движения. А также решена задача по планированию траектории движения схвата манипулятора. Проведен анализ эффективности встроенных в программный пакет функций и выявлены их недостатки при решении указанных задач.

Ключевые слова

Манипулятор, моделирование, метод Денавита — Хартенберга, прямая и обратная позиционные задачи, кинематика, планирование траектории, Matlab, Robotics Toolbox, Guide

Поступила в редакцию 21.11.2019

© МГТУ им. Н.Э. Баумана, 2019

Введение. Одним из наиболее актуальных разделов современной робототехники является разработка мобильных роботов. В настоящее время достаточно актуальным направлением данной тенденции служит создание технологических роботов вертикального перемещения, способных к передвижению по сложным, криволинейным и неструктурированным поверхностям, которые недоступны для традиционных транспортных систем и человека. Во многих случаях для выполнения различных технологических операций, таких как нанесение защитного покрытия на металлоконструкции или покраска и очистка поверхностей, в качестве полезной нагрузки на данных роботах устанавливают манипуляторы, как показано на рис. 1. Поскольку манипуляторы изначально проектируют с гибкой структурой, они способны выполнять достаточно широкий круг задач, т. е. при возможности замены алгоритма работы манипуляционного робота и его перенастройки робот можно переориентировать на выполнение совершенно иных задач как в течение длительного времени, так и краткосрочно.

Разработка и исследование алгоритмов управления манипуляционными роботами осуществляется в специализированных пакетах прикладных программ, которые применяют для моделирования робототехнических систем. Одним из таких пакетов является *MATLAB Robotics Toolbox*. Данная программа позволяет осуществлять полное моделирование работы манипуляторов, т. е. исследовать

кинематику и динамику роботов, планировать и строить траектории их движения и т. п.

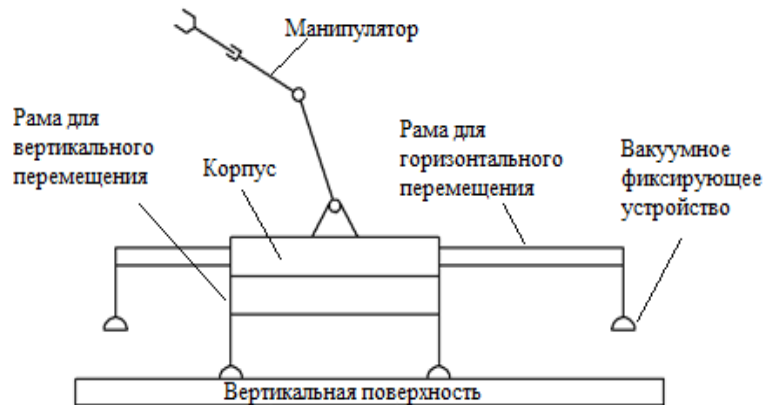


Рис. 1. Мобильный робот с манипулятором

Для разработки алгоритма движения манипулятора прежде всего необходимо изучить его кинематику, т. е. найти пространственное положение манипуляционного робота как функцию времени и соотношение между обобщенными координатами робота и положением его схвата.

Цель данной работы заключается в изучении возможностей работы программного пакета *MATLAB Robotics Toolbox* для исследования кинематики манипулятора. Для исследования наиболее важных аспектов данной темы в работе планируется выполнить ряд задач:

- построение виртуальной модели манипуляционного робота в *MATLAB Robotics Toolbox* для имитации его работы;
- решение прямой и обратной позиционной задач;
- создание приложения с графическим интерфейсом;
- планирование и построение траектории движения схвата манипуляционного робота из заданного начального положения.

Инициализация модели манипулятора. Наиболее распространенным видом промышленного манипуляционного робота является манипулятор с шестью степенями подвижности. Однако вследствие того, что среда, в которой будет осуществляться исследование кинематики робота, а также ее базовые функции имеют некоторые особенности при инициализации модели манипулятора с меньшим количеством степеней подвижности, рассмотрим сначала достаточно простую кинематическую схему манипуляционного робота, а именно трехстепенной манипулятор RRR-типа. Инициализация модели в *MATLAB Robotics Toolbox* будет происходить с помощью задания систем координат звеньев робота методом Денавита — Хартенберга. При описании звена манипулятора используются следующие основные параметры:

- $\theta(i)$ — угол между сочленениями робота, равен углу поворота оси x_{i-1} вокруг оси z_{i-1} до ее совпадения с осью x_i ;

- $d(i)$ — смещение звена, равно расстоянию вдоль оси z_{i-1} от начала $(i - 1)$ -й системы координат до точки пересечения осей z_i и x_i ;
- $a(i)$ — длина звена, определяется как расстояние вдоль оси x_i от z_{i-1} до z_i ;
- $alpha(i)$ — угол поворота звена, равен углу вокруг оси x_i от z_{i-1} до z_i ;
- $sigma(i)$ — тип соединения, равно нулю при вращательном соединении и единице при телескопическом.

Итак, создаем модель манипуляционного робота с параметрами, которые указаны в табл. 1.

Таблица 1

Параметры трехстепенного манипулятора

Номер сочленения	$theta$, град	d , мм	a , мм	$alpha$, град
1	q1	600	0	90
2	q2	0	500	0
3	q3	0	400	0

Для начала последовательно создадим каждое из звеньев манипулятора, используя вектор объектов *Link*:

$$L(i) = Link[theta(i), d(i), a(i), alpha(i), sigma(i)].$$

Теперь соединим звенья в нужной последовательности с помощью класса *SerialLink*. Данный класс представляет собой робота с последовательными связями. Каждая ссылка и соединение в цепочке описывается объектом *Link* с использованием уже упомянутых параметров Денавита — Хартенберга.

```
Robot =
Robot (3 axis, RRR, stdDH, slowRNE)

+---+-----+-----+-----+-----+
| j |   theta |       d |       a |   alpha |
+---+-----+-----+-----+-----+
| 1 |     q1 |   0.6 |     0 |  1.571 |
| 2 |     q2 |     0 |   0.5 |     0 |
| 3 |     q3 |     0 |   0.4 |     0 |
+---+-----+-----+-----+-----+

grav =   0 base = 1 0 0 0 tool = 1 0 0 0
         0       0 1 0 0       0 1 0 0
        9.81     0 0 1 0       0 0 1 0
                0 0 0 1       0 0 0 1
```

Рис. 2. Данные об объекте класса *SerialLink*

Все данные о созданном посредством класса *SerialLink* объекте отображаются в *Command Window* в виде, представленном на рис. 2. О том, что модель манипулятора имеет три звена с вращательными степенями подвижности, свиде-

тельствует признак RRR , а о том, что при описании робота использовалось представление Денавита — Хартенберга, — признак $stdDH$. Помимо этой информации также отображается вектор $grav$, компоненты которого описывают ось, вдоль которой направлена сила тяжести, и две матрицы однородного преобразования размером 4×4 . Они характеризуют положение основания манипулятора и его схвата в базовой системе координат.

Прямая позиционная задача. Прямая позиционная задача кинематики (ППЗ) заключается в нахождении координат положения и ориентации системы координат, связанных с концом кинематической цепи манипулятора, т. е. со схватом или рабочим инструментом, при заданном наборе обобщенных координат. Положение и ориентацию схвата робота в пространстве можно задать при помощи матрицы однородного преобразования размером 4×4 :

$$T = \begin{pmatrix} R & p \\ f & m \end{pmatrix},$$

где R — матрица поворота размером 3×3 ; p — вектор переноса; f — вектор, связанный с вектором центрального проецирования; m — коэффициент масштабирования.

В нашем случае будет использовать матрицу T следующего вида:

$$T = \begin{pmatrix} R & p \\ 000 & 1 \end{pmatrix}.$$

Пусть необходимо найти координаты конца схвата манипуляционного робота в базовой системе координат при следующих обобщенных координатах: $\mathbf{q} = [90^\circ \ 45^\circ \ 0^\circ]^T$.

Решение ППЗ в *MATLAB Robotics Toolbox* осуществляется с помощью функции $fkine(q)$, на вход которой поступает вектор обобщенных координат. Данная команда осуществляет построение матрицы однородного преобразования и выводит её значение в *Command Window*, как показано на рис. 3.

```
T = Robot.fkine([pi/2 pi/4 0])
```

```
T =
```

```
0.0000    -0.0000    1.0000    0.0000
0.7071    -0.7071   -0.0000    0.6364
0.7071     0.7071    0.0000    1.2364
         0         0         0         1.0000
```

Рис. 3. Решение прямой позиционной задачи

Первые три элемента четвертого столбца в матрице T показывают положение рабочего инструмента в базовой системе координат, подматрица R размером 3×3 в левом верхнем углу отображает его ориентацию в пространстве.

Обратная позиционная задача. Обратная позиционная задача (ОПЗ) гораздо чаще применяется на практике, чем прямая. Она заключается в нахождении обобщенных координат всех звеньев при заданном положении и ориентации схвата. В большинстве случаев данная задача имеет несколько решений, т. е. существует более одной конфигурации звеньев манипулятора, при которой могут быть достигнуты желаемое положение и ориентация схвата. Вследствие этого будут существовать множество комбинаций обобщенных координат, являющихся решениями обратной позиционной задачи.

Решается данная задача в *MATLAB Robotics Toolbox* при помощи команды *ikine*(T, q_0, m). На вход данной функции поступают: матрица однородного преобразования T , в которой находится информация о заданном положении и ориентации схвата; q_0 — вектор обобщенных координат, описывающий начальное положение манипулятора; m — вектор размером 1×6 , который используется для блокирования степеней подвижности манипуляторов, в которых их число менее шести. Первые три элемента данного вектора соответствуют положению по осям X , Y и Z , а последние три — вращению вокруг этих осей. Значение элементов в векторе m могут быть равны либо единице, либо нулю (при блокировании). Количество ненулевых элементов должно равняться числу степеней подвижности манипулятора. В результате применения команды *ikine* в *Command Window* выводится решение обратной позиционной задачи в виде, показанном на рис. 4.

```
q = Robot.ikine(T, [0 0 0], [1 1 1 0 0 0])
q =
-1.5708    2.3556    0.0013
```

Рис. 4. Решение обратной позиционной задачи

Существенным недостатком данной функции является то, что она не позволяет учитывать неоднозначность решения ОПЗ.

Создание приложения с графическим интерфейсом в среде GUIDE. Для более удобного и наглядного исследования кинематики манипуляционного робота целесообразно создать приложение с графическим интерфейсом. Воспользуемся для этой цели внутренней средой *MATLAB* под названием *GUIDE* (рис. 5).

Для ввода исходных данных, при которых ищутся решения прямой и обратной позиционной задач, будем использовать элементы *Edit Text*. Для запуска поиска решения данных задач воспользуемся элементами *Push Button*, а для отрисовки модели манипулятора в заданном положении добавим также элемент *Axes*. Итак, при решении прямой позиционной задачи в поля элементов *Edit Text* в первом столбце вводятся исходные обобщенных координаты звеньев

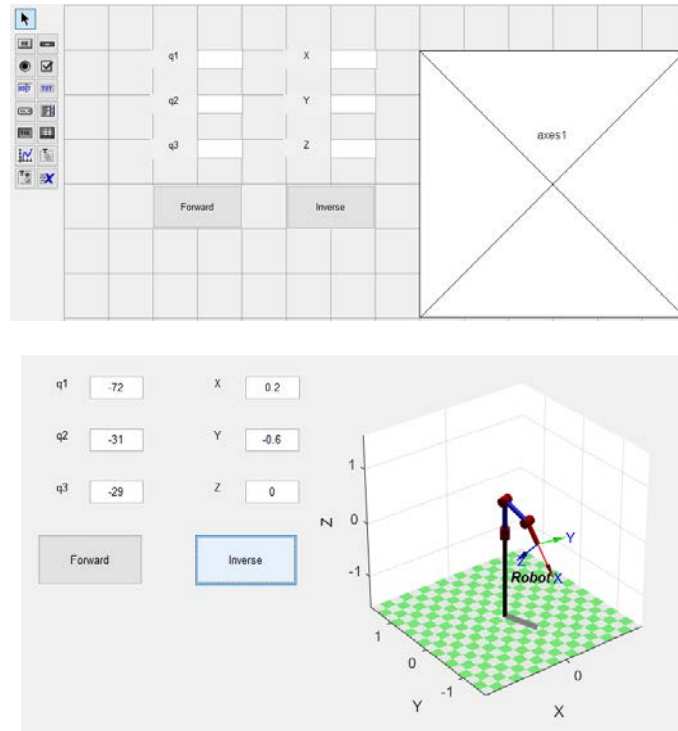


Рис. 5. Среда GUIDE

манипулятора. По нажатию кнопки *Forward* запускается команда по поиску решения ППЗ, вследствие чего в полях элементов *Edit Text* во втором столбце будут выведены найденные решения. Код программы, который реализует все описанные выше действия, представлен на рис. 6.

```
function btn_Forward_Callback(hObject, eventdata, handles)
Th_1 = str2double(get(handles.Theta_1,'String'))*pi/180;
Th_2 = str2double(get(handles.Theta_2,'String'))*pi/180;
Th_3 = str2double(get(handles.Theta_3,'String'))*pi/180;

L_1 = 60;
L_2 = 50;
L_3 = 40;

L(1) = Link([0 L_1 0 pi/2]);
L(2) = Link([0 0 L_2 0]);
L(3) = Link([0 0 L_3 0]);

Robot = SerialLink(L);
Robot.name = 'Robot';
Robot.plot([Th_1 Th_2 Th_3]);

T = Robot.fkine([Th_1 Th_2 Th_3]);
handles.Pos_X.String = num2str(floor(T(1,4)));
handles.Pos_Y.String = num2str(floor(T(2,4)));
handles.Pos_Z.String = num2str(floor(T(3,4)));
```

Рис. 6. Код программы

Аналогично, при решении ОПЗ во второй столбец вводятся исходные координаты рабочего инструмента манипуляционного робота и при нажатии кнопки *Inverse* запускается команда, которая отображает найденное решение ОПЗ в первом столбце.

Планирование траектории движения манипулятора. Прежде чем учитывать динамические характеристики манипуляционного робота, необходимо в полной мере рассмотреть кинематический уровень, который обеспечивает построение программных движений приводов, размещенных в сочленениях манипулятора, и соответственно решить задачу кинематического управления. В свою очередь, данная задача подразделяется на два этапа:

- 1) планирование траектории, т. е. предварительное определение движения степеней подвижности $q(t)$ на некотором временном интервале;
- 2) отработка полученной траектории движения непосредственно приводами сочленений.

Далее рассмотрим первый этап задачи кинематического управления — планирование траектории движения в пространстве обобщенных координат, которая будет обеспечивать плавное перемещение схвата манипулятора из заданного начального положения q_0 в конечное q_e за отрезок времени $[t_0, t_e]$ со следующими граничными условиями:

$$\begin{aligned} q(t_0) &= q_0; \\ q(t_e) &= q_e; \\ \dot{q}(t_0) &= \dot{q}(t_e) = 0; \\ \ddot{q}(t_0) &= \ddot{q}(t_e) = 0. \end{aligned}$$

Планирование траектории будем осуществлять для реальной модели шестистепенного манипулятора, параметры которого указаны в табл. 2.

Таблица 2

Параметры шестистепенного манипулятора

Номер сочленения	$theta$, град	d , мм	a , мм	$alpha$, град
1	q_1	300	100	90
2	q_2	0	800	0
3	q_3	0	0	90
4	q_4	743	0	-90
5	q_5	0	0	90
6	q_6	200	0	90

Существует несколько подходов к построению траектории движения, переводящей рабочий орган манипулятора из q_0 в q_e и удовлетворяющей условиям гладкости. Наиболее распространенными из них являются применение специальной параметризации или использование полиномиальной интерполяции. Воспользу-

емся последним из методов. Полиномом с минимальной степенью, который удовлетворяет условиям гладкости, является многочлен пятой степени:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5.$$

Первая и вторая производные от данного полинома также являются гладкими функциями:

$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4;$$

$$\ddot{q}(t) = 2a_2 + 6a_3 t + 4a_4 t^2 + 5a_5 t^3.$$

Пусть $t_0 = 0$, тогда систему уравнений для поиска коэффициентов исходного полинома пятой степени можно составить следующим образом:

$$\begin{cases} a_0 = q_0; \\ a_1 = 0; \\ a_2 = 0; \\ a_0 + a_1 t_e + a_2 t_e^2 + a_3 t_e^3 + a_4 t_e^4 + a_5 t_e^5 = q_e; \\ a_1 + 2a_2 t_e + 3a_3 t_e^2 + 4a_4 t_e^3 + 5a_5 t_e^4 = 0; \\ 2a_2 + 6a_3 t_e + 12a_4 t_e^2 + 20a_5 t_e^3 = 0. \end{cases}$$

В программном пакете *MATLAB Robotics Toolbox* планирование и построение траектории движения между двумя точками пространства осуществляется с помощью функции *jtraj(q₀, q_e, m)* внутри класса *SerialLink*. Функция *jtraj* использует полином пятого порядка с нулевыми граничными условиями по умолчанию для скорости и ускорения. Ее параметры включают в себя матрицы однородного преобразования, описывающие начальное и конечное положение схвата манипулятора, а также число шагов по времени. На выходе получаем матрицу размером $m \times N$, где m — число шагов по времени, а N — число звеньев манипулятора, в нашем случае $N = 6$. Кроме того, данная функция позволяет по желанию пользователя выводить матрицы скорости и ускорения таких же размеров. Траектория движения робота, построенная с использованием функции *jtraj*, показана на рис. 7.

Заключение. *MATLAB Robotics Toolbox* является достаточно практичным и эффективным программным пакетом для моделирования манипуляционных роботов. Более того, благодаря большой библиотеке встроенных классов и функций приложение является достаточно простым для освоения рядовыми пользователями, а также помогает охватить достаточно большой круг задач. Несмотря на то что в данной работе были рассмотрены лишь некоторые из основных задач робототехники, *MATLAB Robotics Toolbox* позволяет решить их с достаточно высокой скоростью и легкостью. Помимо исследования кинематики,

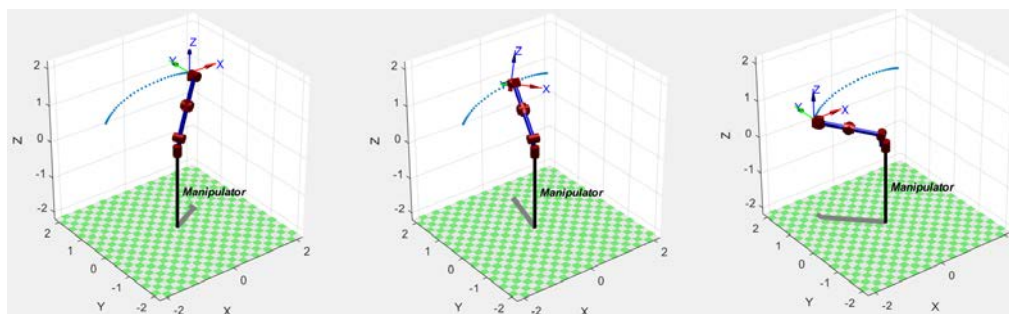


Рис. 7. Движение манипулятора из начального положения в конечное по траектории, построенной функцией *jtraj*

которое непосредственно проводилось в данной работе, программа также дает возможность изучить влияние динамических характеристик. Ещё одним неоспоримым преимуществом программы является то, что ее исходные коды находятся в открытом доступе, что позволяет пользователям постоянно их оптимизировать и улучшать. Однако были выявлены и некоторые недостатки данного пакета, например, невозможность вывода нескольких решений обратной позиционной задачи при ее неоднозначности. Таким образом, программный *MATLAB Robotics Toolbox* может быть использован как для обучения, так и для моделирования работы реальных роботов.

Литература

- [1] Зенкевич С.Л., Ющенко А.С. Управление роботами. М., Изд-во МГТУ им. Н.Э. Баумана, 2000.
- [2] Макаров И.М., ред. Робототехника и гибкие автоматизированные производства. Кн. 5. Моделирование робототехнических систем и гибких автоматизированных производств. М., Высшая школа, 1986.
- [3] Борисов О.И., Громов В.С., Пыркин А.А. Методы управления робототехническими приложениями. СПб., ИТМО, 2016.
- [4] Градецкий В.Г., Вешников В.Б., Калиниченко С.В. и др. Управляемое движение мобильных роботов по произвольно ориентированным в пространстве поверхностям. М., Наука, 2001.
- [5] Чемоданов Б.К., ред. Математические основы теории автоматического регулирования. М., Высшая школа, 1977.
- [6] Градецкий В.Г., Рачков М.Ю. Роботы вертикального перемещения. М., Минобрнауки РФ, 1997.
- [7] Ким Д.П. Теория автоматического управления. М., Физматлит, 2003.
- [8] Corke P. Robotics toolbox. *petercorke.com: веб-сайт*. URL: <http://petercorke.com/wordpress/toolboxes/robotics-toolbox> (дата обращения: 14.10.2019).
- [9] Corke P. Robotics, vision and control: fundamental algorithms in MATLAB. *petercorke.com: веб-сайт*. URL: <http://www.petercorke.com/RVC1/> (дата обращения: 14.10.2019).
- [10] Panchal K., Vyas C., Patel D. Developing the prototype of wall climbing robot. *IJAERD*, 2014, vol. 1, no. 3, pp. 58–65.

- [11] Shmidt D., Berns K. Climbing robots for maintenance and inspections of vertical structures – A survey of design aspects and technologies. *Robot. Auton. Syst.*, 2013, vol. 61, no. 12, pp. 1288–1305. DOI: <http://dx.doi.org/10.1016/j.robot.2013.09.002>

Егоров Евгений Евгеньевич — студент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Калиниченко Сергей Владимирович, кандидат технических наук, доцент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:

Егоров Е.Е. Моделирование работы манипуляционного робота в программном пакете Matlab Robotics Toolbox. *Политехнический молодежный журнал*, 2020, № 01(42). <http://dx.doi.org/10.18698/2541-8009-2020-01-567>

SIMULATING THE MANIPULATION ROBOT OPERATION IN THE MATLAB ROBOTICS TOOLBOX SOFTWARE

E.E. Egorov

egorov.18011996@gmail.com

SPIN-code: 2333-2610

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The paper investigates the capabilities of the MATLAB Robotics Toolbox software package for modeling the movement of a manipulation robot. An algorithm is described for creating a manipulator model in a program based on the Denavit-Hartenberg representation. The study is presented of the manipulation robot kinematics. The direct and inverse positional problems are solved using the internal functions of the software package. The creation of a GUI application in the GUIDE environment for rendering a robot model and modeling its movement is considered. Also authors solved the problem of planning the trajectory of the manipulator grip. The analysis is carried out of the effectiveness of the functions built into the software package and their shortcomings in solving these problems are identified.

Keywords

Manipulator, simulation, Denavit-Hartenberg method, direct and inverse positional problems, kinematics, path planning, Matlab, Robotics Toolbox, Guide

Received 21.11.2019

© Bauman Moscow State Technical University, 2019

References

- [1] Zenkevich S.L., Yushchenko A.S. Upravlenie robotami [Robot guidance]. Moscow, Bauman MSTU Publ., 2000 (in Russ.).
- [2] Makarov I.M., ed. Robototekhnika i gibkie avtomatizirovannye proizvodstva. Kn. 5. Modelirovanie robototekhnicheskikh sistem i gibkikh avtomatizirovannykh proizvodstv [Robotics and flexible manufacturing systems. Vol. 5. Simulation of robotic systems and flexible manufacturing]. Moscow, Vysshaya shkola Publ., 1986 (in Russ.).
- [3] Borisov O.I., Gromov V.S., Pyrkin A.A. Metody upravleniya robototekhnicheskimi prilozheniyami [Guidance methods for robotic applications]. Sankt-Petersburg, ITMO Publ., 2016 (in Russ.).
- [4] Gradetskiy V.G., Veshnikov V.B., Kalinichenko S.V., et al. Upravlyaemoe dvizhenie mobil'nykh robotov po proizvol'no orientirovannym v prostranstve poverkhnostyam [guided motion of mobile robots on random spatially-oriented surfaces]. Moscow, Nauka Publ., 2001 (in Russ.).
- [5] Chemodanov B.K., ed. Matematicheskie osnovy teorii avtomaticheskogo regulirovaniya [Mathematical fundamentals of automatic regulation theory]. Moscow, Vysshaya shkola Publ., 1977 (in Russ.).
- [6] Gradetskiy V.G., Rachkov M.Yu. Roboty vertikal'nogo peremeshcheniya [Vertical motion robots]. Moscow, Minobrazovaniya RF Publ., 1997 (in Russ.).
- [7] Kim D.P. Teoriya avtomaticheskogo upravleniya [Automatic control theory]. Moscow, Fizmatlit Publ., 2003 (in Russ.).

- [8] Corke P. Robotics toolbox. *petercorke.com: website*. URL: <http://petercorke.com/wordpress/toolboxes/robotics-toolbox> (accessed: 14.10.2019).
- [9] Corke P. Robotics, vision and control: fundamental algorithms in MATLAB. *petercorke.com: website*. URL: <http://www.petercorke.com/RVC1/> (accessed: 14.10.2019).
- [10] Panchal K., Vyas C., Patel D. Developing the prototype of wall climbing robot. *IJAERD*, 2014, vol. 1, no. 3, pp. 58–65.
- [11] Shmidt D., Berns K. Climbing robots for maintenance and inspections of vertical structures – A survey of design aspects and technologies. *Robot. Auton. Syst.*, 2013, vol. 61, no. 12, pp. 1288–1305. DOI: <http://dx.doi.org/10.1016/j.robot.2013.09.002>

Egorov E.E. — Student, Department of Robotics and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Kalinichenko S.V., Cand. Sc. (Eng.), Assoc. Professor, Department of Robotics and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:

Egorov E.E. Simulating the manipulation robot operation in the Matlab Robotics Toolbox software. *Politekhničeskij molodezhnyy zhurnal* [Politechnical student journal], 2020, no. 01(42). <http://dx.doi.org/10.18698/2541-8009-2020-01-567.html> (in Russ.).